In [1]:
```
# Read The Data From kaggle Website
# Source  ------------------------->>>>      https://www.kaggle.com/c/m5-forecast
#!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0
```

In [2]:
```
# Read The Price Preprocessed Data   <<<---  Price_Data_Preprocessing_1.pkl   --

#!wget --header="Host: doc-04-0c-docs.googleusercontent.com" --header="User-Agent
```

In [1]:
```
# Read The Calander Data   <<<---  Calander_Data_Preprocessing_2.pkl   --->>>

#!wget --header="Host: doc-08-0c-docs.googleusercontent.com" --header="User-Agent
```

In [4]:
```
#import All The Library

import pandas as pd
import numpy as np
```

Part- A. Read The Data

A_1. Read Price Data

In [5]:
```
Price = pd.read_pickle("Price_Data_Preprocessing_1.pkl")
Price.head()
```

Out[5]:

	store_id	item_id	wm_yr_wk	sell_price	price_diffrence_2	price_diffrence_4	price_d
368746	CA_1	FOODS_1_001	11101	2.0	NaN	NaN	
368747	CA_1	FOODS_1_001	11102	2.0	0.0	NaN	
368748	CA_1	FOODS_1_001	11103	2.0	0.0	NaN	
368749	CA_1	FOODS_1_001	11104	2.0	0.0	0.0	
368750	CA_1	FOODS_1_001	11105	2.0	0.0	0.0	

In [6]:
```
Price.shape
```

Out[6]:
```
(6841121, 7)
```

A_2. Read Calander Data

In [7]:
```
cal = pd.read_pickle("Calander_Data_Preprocessing_2.pkl")
cal.head()
```

Out[7]:

	date	wm_yr_wk	weekday	wday	month	year	d	event_type_1	event_type_2	snap_CA
0	2011-01-29	11101	Saturday	1	1	2011	d_1	NaN	NaN	0
1	2011-01-30	11101	Sunday	2	1	2011	d_2	NaN	NaN	0
2	2011-01-31	11101	Monday	3	1	2011	d_3	NaN	NaN	0
3	2011-02-01	11101	Tuesday	4	2	2011	d_4	NaN	NaN	1
4	2011-02-02	11101	Wednesday	5	2	2011	d_5	NaN	NaN	1

In [8]:
```
# Devide Each Months Into Six Groups To Caputre Seasonality if it's There

dict_day_to_group = {
    1:1, 2:1, 3:1, 4:1, 5:1,
    6:2, 7:2, 8:2, 9:2, 10:2,
    11:3, 12:3, 13:3, 14:3, 15:3,
    16:4, 17:4, 18:4, 19:4, 20:4,
    21:5, 22:5, 23:5, 24:5, 25:5,
    26:6, 27:6, 28:6, 29:6, 30:6, 31:6
}

cal['group_day'] = cal['date'].str[-2:].apply(int).map(dict_day_to_group)
```

In [9]:
```
cal.shape
```

Out[9]:
```
(1969, 27)
```

A_3. Read Sales Data

In [10]:
```
# Unzip the Data

#!unzip m5-forecasting-accuracy.zip
```

In [11]:
```
# Beacuse of Memory Constarint i am Skipping First 1049 Days of Sales

first_day = 1050                # Training Days   -- 1050 to 1913 Days      (865
last_day = 1941                 # Validation Days -- 1914 to 1941 Days      (28 
                                # Test Days       --  1942 to 1969 Days      (28 
```

In [12]:
```
# Numerical Columns

numcols = [f"d_{day}" for day in range(first_day,last_day+1)]
print("First 5 Days ",numcols[0:5])
print("Last 5 Days ",numcols[-5:])
```

```
First 5 Days  ['d_1050', 'd_1051', 'd_1052', 'd_1053', 'd_1054']
Last 5 Days  ['d_1937', 'd_1938', 'd_1939', 'd_1940', 'd_1941']
```

In [13]:
```
# Categorical Columns

catcols = ['id', 'item_id', 'dept_id','store_id', 'cat_id', 'state_id']
```

In [14]:
```
# data Types

dtype = {numcol:"float32" for numcol in numcols}
dtype.update({col: "category" for col in catcols if col != "id"})
```

In [15]:
```
# Read The Sales data

sales = pd.read_csv("sales_train_evaluation.csv", usecols = catcols + numcols, d
sales.head()
```

Out[15]:

	id	item_id	dept_id	cat_id	store_id	state_id	d_1
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	

5 rows × 898 columns

In [16]:
```
sales.shape
```

Out[16]:
```
(30490, 898)
```

In [17]:
```
# Add Test Data

for day in range(1942,1970):
  sales['d_' + str(day)] = np.nan
```

In [18]:
```
sales.head()
```

Out[18]:

	id	item_id	dept_id	cat_id	store_id	state_id	d_1(
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	

5 rows × 926 columns

Part- B. Downcast The Data

B_1. Downcasting the Data

In [19]:
```
#https://www.kaggle.com/anshuls235/time-series-forecasting-eda-fe-modelling/notel

def downcast(df):

    cols = df.dtypes.index.tolist()
    types = df.dtypes.values.tolist()

    for i,t in enumerate(types):

        if 'int' in str(t):
            if df[cols[i]].min() > np.iinfo(np.int8).min and df[cols[i]].max() <
                df[cols[i]] = df[cols[i]].astype(np.int8)
            elif df[cols[i]].min() > np.iinfo(np.int16).min and df[cols[i]].max(
                df[cols[i]] = df[cols[i]].astype(np.int16)
            elif df[cols[i]].min() > np.iinfo(np.int32).min and df[cols[i]].max(
                df[cols[i]] = df[cols[i]].astype(np.int32)
            else:
                df[cols[i]] = df[cols[i]].astype(np.int64)

        elif 'float' in str(t):
            if df[cols[i]].min() > np.finfo(np.float16).min and df[cols[i]].max(
                df[cols[i]] = df[cols[i]].astype(np.float16)
            elif df[cols[i]].min() > np.finfo(np.float32).min and df[cols[i]].max
                df[cols[i]] = df[cols[i]].astype(np.float32)
            else:
                df[cols[i]] = df[cols[i]].astype(np.float64)

        elif t == np.object:
            if cols[i] == 'date':
                df[cols[i]] = pd.to_datetime(df[cols[i]], format='%Y-%m-%d')
            else:
                df[cols[i]] = df[cols[i]].astype('category')

    return df
```

In [20]:
```
Price = downcast(Price)
cal = downcast(cal)
sales = downcast(sales)
```

Part- C. Label Encoding

C_1. Sales DataFrame Label Encoding

In [21]:
```
sales.head()
```

Out[21]:

	id	item_id	dept_id	cat_id	store_id	state_id	d_1
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	

5 rows × 926 columns

In [22]:
```
# Using label Encoding

for col in catcols:
    if col != "id":
        sales[col] = sales[col].cat.codes.astype("int16")
        sales[col] -= sales[col].min()
```

In [23]:
```
sales.tail()
```

Out[23]:

	id	item_id	dept_id	cat_id	store_id	state_id	d_1050	d_1051
30485	FOODS_3_823_WI_3_evaluation	3044	6	2	9	2	2.0	1.0
30486	FOODS_3_824_WI_3_evaluation	3045	6	2	9	2	0.0	0.0
30487	FOODS_3_825_WI_3_evaluation	3046	6	2	9	2	0.0	0.0
30488	FOODS_3_826_WI_3_evaluation	3047	6	2	9	2	0.0	3.0
30489	FOODS_3_827_WI_3_evaluation	3048	6	2	9	2	0.0	0.0

5 rows × 926 columns

C_2. Calander DataFrame Label Encoding

In [24]:
```
cal.tail()
```

Out[24]:

	date	wm_yr_wk	weekday	wday	month	year	d	event_type_1	event_type_2	sna
1964	2016-06-15	11620	Wednesday	5	6	2016	d_1965	NaN	NaN	
1965	2016-06-16	11620	Thursday	6	6	2016	d_1966	NaN	NaN	
1966	2016-06-17	11620	Friday	7	6	2016	d_1967	NaN	NaN	
1967	2016-06-18	11621	Saturday	1	6	2016	d_1968	NaN	NaN	
1968	2016-06-19	11621	Sunday	2	6	2016	d_1969	Sporting	Cultural	

In [25]:
```
cal.dtypes
```

Out[25]:
```
date                          datetime64[ns]
wm_yr_wk                               int16
weekday                             category
wday                                   int8
month                                  int8
year                                  int16
d                                   category
event_type_1                        category
event_type_2                        category
snap_CA                                int8
snap_TX                                int8
snap_WI                                int8
event_name_1_Cinco De Mayo            float16
event_name_1_Father's day             float16
event_name_1_MemorialDay              float16
event_name_1_Mother's day             float16
event_name_1_NBAFinalsEnd             float16
event_name_1_NBAFinalsStart           float16
event_name_1_OrthodoxEaster           float16
event_name_1_Pesach End               float16
event_name_1_Ramadan starts           float16
event_name_2_Cinco De Mayo            float16
event_name_2_Father's day             float16
event_name_2_OrthodoxEaster           float16
Event_1                             category
Event_2                             category
group_day                              int8
dtype: object
```

In [26]:
```
Cal_DTYPES = {"event_type_1": "category", "event_type_2": "category","weekday":
              "Event_1":"category","Event_2":"category"}
```

In [27]:
```
# Apply Label Encoding
for col, col_dtype in Cal_DTYPES.items():
        cal[col] = cal[col].cat.codes.astype("int16")
        cal[col] -= cal[col].min()
```

In [28]:
```
cal.tail()
```

Out[28]:

	date	wm_yr_wk	weekday	wday	month	year	d	event_type_1	event_type_2	snap_
1964	2016-06-15	11620	6	5	6	2016	d_1965	0	0	
1965	2016-06-16	11620	4	6	6	2016	d_1966	0	0	
1966	2016-06-17	11620	0	7	6	2016	d_1967	0	0	
1967	2016-06-18	11621	2	1	6	2016	d_1968	0	0	
1968	2016-06-19	11621	3	2	6	2016	d_1969	4	1	

C_3. Price DataFrame Label Encoding

In [29]:
```
Price.head()
```

Out[29]:

	store_id	item_id	wm_yr_wk	sell_price	price_diffrence_2	price_diffrence_4	price_d
368746	CA_1	FOODS_1_001	11101	2.0	NaN	NaN	
368747	CA_1	FOODS_1_001	11102	2.0	0.0	NaN	
368748	CA_1	FOODS_1_001	11103	2.0	0.0	NaN	
368749	CA_1	FOODS_1_001	11104	2.0	0.0	0.0	
368750	CA_1	FOODS_1_001	11105	2.0	0.0	0.0	

In [30]:
```
Price.dtypes
```

Out[30]:
```
store_id             category
item_id              category
wm_yr_wk                int16
sell_price            float16
price_diffrence_2     float16
price_diffrence_4     float16
price_diffrence_8     float16
dtype: object
```

In [31]:
```
PRICE_DTYPES = {"store_id": "category", "item_id": "category" }
```

In [32]:
```
# Apply Label Encoding
for col, col_dtype in PRICE_DTYPES.items():
        Price[col] = Price[col].cat.codes.astype("int16")
        Price[col] -= Price[col].min()
```

In [33]:
```
Price.head()
```

Out[33]:

	store_id	item_id	wm_yr_wk	sell_price	price_diffrence_2	price_diffrence_4	price_diffrenc
368746	0	0	11101	2.0	NaN	NaN	N
368747	0	0	11102	2.0	0.0	NaN	N
368748	0	0	11103	2.0	0.0	NaN	N
368749	0	0	11104	2.0	0.0	0.0	N
368750	0	0	11105	2.0	0.0	0.0	N

D. Create Final Dataset

D_1. Join Sales And Calander Data

In [34]:
```
sales.head()
```

Out[34]:

	id	item_id	dept_id	cat_id	store_id	state_id	d_1050	d_1051	d_
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	0.0	0.0	
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	0.0	2.0	
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	0.0	0.0	
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	0.0	1.0	
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	0.0	3.0	

5 rows × 926 columns

In [35]:
```
# Convert Sales Data From Wide Fromat to Long Format

Sales_L = pd.melt(sales,
                  id_vars=[a for a in sales.columns if a.find("id")!=-1],      #
                  value_vars=[a for a in sales.columns if a.find("d_")==0],    #
                  var_name='d',
                  value_name='sales')

Sales_L.head()
```

Out[35]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	d_1050	0.0
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	d_1050	0.0
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	d_1050	0.0
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	d_1050	0.0
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	d_1050	0.0

In [36]:
```
cal.head()
```

Out[36]:

	date	wm_yr_wk	weekday	wday	month	year	d	event_type_1	event_type_2	snap_CA	sn
0	2011-01-29	11101	2	1	1	2011	d_1	0	0	0	
1	2011-01-30	11101	3	2	1	2011	d_2	0	0	0	
2	2011-01-31	11101	1	3	1	2011	d_3	0	0	0	
3	2011-02-01	11101	5	4	2	2011	d_4	0	0	1	
4	2011-02-02	11101	6	5	2	2011	d_5	0	0	1	

In [37]:
```
# join Both DataFrame Using Same Columns

Sales_L = Sales_L.merge(cal, on= "d", copy = False)
Sales_L.head()
```

Out[37]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	da
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	d_1050	0.0	201 12-1
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	d_1050	0.0	201 12-1
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	d_1050	0.0	201 12-1
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	d_1050	0.0	201 12-1
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	d_1050	0.0	201 12-1

In [38]:
```
Sales_L.shape
```

Out[38]: (28050800, 34)

D_2. Join Sales And Price Data

In [39]:
```
Sales_L = Sales_L.merge(Price, on = ["store_id", "item_id", "wm_yr_wk"], copy = F
Sales_L.head()
```

Out[39]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	da
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	d_1050	0.0	201 12-1
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	d_1050	0.0	201 12-1
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	d_1050	0.0	201 12-1
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	d_1050	0.0	201 12-1
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	d_1050	0.0	201 12-1

In [40]:
```
Sales_L.tail()
```

Out[40]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales
27023816	FOODS_3_825_WI_3_evaluation	3046	6	2	9	2	d_1969	NaN
27023817	FOODS_3_826_WI_3_evaluation	3047	6	2	9	2	d_1968	NaN
27023818	FOODS_3_826_WI_3_evaluation	3047	6	2	9	2	d_1969	NaN
27023819	FOODS_3_827_WI_3_evaluation	3048	6	2	9	2	d_1968	NaN
27023820	FOODS_3_827_WI_3_evaluation	3048	6	2	9	2	d_1969	NaN

In [41]:
```
Sales_L.shape
```

Out[41]:
```
(27023821, 38)
```

In [42]:
```
Sales_L = downcast(Sales_L)
```

E. Lag Features

In [43]:
```
%%time
# Lag 7 and Lag 28

Sales_L['lag_7'] = Sales_L[["id","sales"]].groupby("id")["sales"].shift(7)
Sales_L['lag_28'] = Sales_L[["id","sales"]].groupby("id")["sales"].shift(28)
```

```
CPU times: user 2.7 s, sys: 1.79 s, total: 4.49 s
Wall time: 4.49 s
```

In [44]:
```
Sales_L.head()
```

Out[44]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	da
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	d_1050	0.0	201 12-1
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	d_1050	0.0	201 12-1
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	d_1050	0.0	201 12-1
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	d_1050	0.0	201 12-1
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	d_1050	0.0	201 12-1

F. Rolling Mean Features

In [45]:
```
# Time Consuming Process--- Be Patient
%%time

Sales_L['rmean_7_7'] = Sales_L[["id", 'lag_7']].groupby("id")['lag_7'].transform
```

```
CPU times: user 15min 48s, sys: 4.25 s, total: 15min 52s
Wall time: 15min 53s
```

In [46]:
```
%%time

Sales_L['rmean_7_28'] = Sales_L[["id", 'lag_7']].groupby("id")['lag_7'].transfor
```

```
CPU times: user 17min 25s, sys: 19.6 s, total: 17min 45s
Wall time: 17min 45s
```

In [47]:
```
%%time

Sales_L['rmean_28_7'] = Sales_L[["id", 'lag_28']].groupby("id")['lag_28'].transf
```

```
CPU times: user 17min 27s, sys: 13.4 s, total: 17min 41s
Wall time: 17min 41s
```

In [48]:
```
%%time

Sales_L['rmean_28_28'] = Sales_L[["id", 'lag_28']].groupby("id")['lag_28'].trans
```

```
CPU times: user 17min 19s, sys: 1min 8s, total: 18min 28s
Wall time: 18min 28s
```

G. Date Related Feature

In [49]:
```
#https://www.programiz.com/python-programming/methods/built-in/getattr

date_features = {

    "wday": "weekday",
    "week": "weekofyear",
    "month": "month",
    "quarter": "quarter",
    "year": "year",
    "mday": "day"
}
```

In [50]:
```
for date_feat_name, date_feat_func in date_features.items():
    if date_feat_name in Sales_L.columns:
        # If Feature Present in my DataFrame than only Change DataType
        Sales_L[date_feat_name] = Sales_L[date_feat_name].astype("int16")
    else:
        # If Feature is not Present in my DataFrame than Get The Feature
        Sales_L[date_feat_name] = getattr(Sales_L["date"].dt, date_feat_func).as
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: FutureWarning:
Series.dt.weekofyear and Series.dt.week have been deprecated.  Please use Serie
s.dt.isocalendar().week instead.
  import sys
```

In [51]:
```
Sales_L.head()
```

Out[51]:

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales	da
0	HOBBIES_1_001_CA_1_evaluation	0	0	0	0	0	d_1050	0.0	201 12-1
1	HOBBIES_1_002_CA_1_evaluation	1	0	0	0	0	d_1050	0.0	201 12-1
2	HOBBIES_1_003_CA_1_evaluation	2	0	0	0	0	d_1050	0.0	201 12-1
3	HOBBIES_1_004_CA_1_evaluation	3	0	0	0	0	d_1050	0.0	201 12-1
4	HOBBIES_1_005_CA_1_evaluation	4	0	0	0	0	d_1050	0.0	201 12-1

In [52]:
```
Sales_L.shape
```

Out[52]:
```
(27023821, 47)
```

In [53]:
```
Sales_L = downcast(Sales_L)
```

In [54]:
```
Sales_L.dtypes
```

Out[54]:
```
id                                category
item_id                              int16
dept_id                               int8
cat_id                                int8
store_id                              int8
state_id                              int8
d                                 category
sales                              float16
date                         datetime64[ns]
wm_yr_wk                             int16
weekday                               int8
wday                                  int8
month                                 int8
year                                 int16
event_type_1                          int8
event_type_2                          int8
snap_CA                               int8
snap_TX                               int8
snap_WI                               int8
event_name_1_Cinco De Mayo         float16
event_name_1_Father's day          float16
event_name_1_MemorialDay           float16
event_name_1_Mother's day          float16
event_name_1_NBAFinalsEnd          float16
event_name_1_NBAFinalsStart        float16
event_name_1_OrthodoxEaster        float16
event_name_1_Pesach End            float16
event_name_1_Ramadan starts        float16
event_name_2_Cinco De Mayo         float16
event_name_2_Father's day          float16
event_name_2_OrthodoxEaster        float16
Event_1                               int8
Event_2                               int8
group_day                             int8
sell_price                         float16
price_diffrence_2                  float16
price_diffrence_4                  float16
price_diffrence_8                  float16
lag_7                              float16
lag_28                             float16
rmean_7_7                          float16
rmean_7_28                         float16
rmean_28_7                         float16
rmean_28_28                        float16
week                                  int8
quarter                               int8
mday                                  int8
dtype: object
```

In [55]:
```
Sales_L.shape
```

Out[55]:
```
(27023821, 47)
```

In [56]:
```
# Save All The Data in Google Drive

from google.colab import  drive
drive.mount('/drive')
```

```
Mounted at /drive
```

In [57]:
```
Sales_L.to_pickle('/drive/My Drive/Case_Study1/Data_m5.pkl')
```